QP-Spline-ST-Speed Optimizer

1 Definition

After finding a path in QP-Spline-Path, Apollo converts all obstacles on the path and the ADV (autonomous driving vehicle) into an ST graph, which represents the station changes over time along the path. The speed optimization task is to find a path on the ST graph that is collision-free and safe.

Apollo uses spline to define the path. To find the best path, Apollo leverages Quadratic programming with a set of conditions. The QP formulation is defined as:

$$\frac{1}{2} \cdot x^T \cdot H \cdot x + f^T \cdot x$$
$$s.t. \, LB \leq x \leq UB$$
$$A_{eq} x = b_{eq}$$
$$Ax \leq b$$

2 Objective function

1.1 Get spline segments

Split the path into **n** segments. Each segment trajectory is defined by a polynomial.

1.2 Define function for each spline segment

Each segment ***i*** has accumulated distance d_i along a reference line. And the trajectory for the segment is defined as a polynomial of degree five by default.

$$s = f_i(t) = a_{0i} + a_{1i} \cdot t + a_{2i} \cdot t^2 + a_{3i} \cdot t^3 + a_{4i} \cdot t^4 + a_{5i} \cdot t^5$$

1.3 Define objective function of optimization for each segment

Apollo first defines $cost_1$ to make the trajectory smooth:

$$cost_1 = \sum_{i=1}^{n} \left(w_1 \cdot \int_0^{d_i} (f_i')^2(s) ds + w_2 \cdot \int_0^{d_i} (f_i'')^2(s) ds + w_3 \cdot \int_0^{d_i} (f_i''')^2(s) ds \right)$$

Then Apollo defines $cost_2$ as the difference between the final ST trajectory and the cruise ST trajectory (with given speed limits — m points):

$$cost_2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(f_i(t_j) - s_j \right)^2$$

Similarly, Apollo defines $cost_3$ that is the difference between the first ST path and the follow ST path (o points):

$$cost_3 = \sum_{i=1}^{n} \sum_{j=1}^{o} \left(f_i(t_j) - s_j \right)^2$$

Finally, the objective function is defined as:

$$cost = cost_1 + cost_2 + cost_3$$

3 Constraints

3.1 The init point constraints

Given the assumption that the the first point is ($t0$, $s0$), and $s0$ is on the planned path $f_i(t)$, $f'i(t)$, and $f_i(t)''$ (position, velocity, acceleration). Apollo converts those constraint into QP equality constraints:

$$A_{eq}x = b_{eq}$$

3.2 Monotone constraint

The path must be monotone, e.g., the vehicle can only drive forward.

Sample **m** points on the path, for each j and $j-1$ point pairs ($j \in [1, \dots, m]$):

If the two points on the same spline k:

$$\begin{vmatrix} 1 & t_j & t_j^2 & t_j^3 & t_j^4 & t_j^5 \end{vmatrix} \cdot \begin{vmatrix} a_k \\ b_k \\ c_k \\ d_k \\ e_k \\ f_k \end{vmatrix} > \begin{vmatrix} 1 & t_{j-1} & t_{j-1}^2 & t_{j-1}^3 & t_{j-1}^4 & t_{j-1}^5 \end{vmatrix} \cdot \begin{vmatrix} a_k \\ b_k \\ c_k \\ d_k \\ e_k \\ f_k \end{vmatrix}$$

If the two points on the different spline k and l:

$$\begin{vmatrix} 1 & t_j & t_j^2 & t_j^3 & t_j^4 & t_j^5 \end{vmatrix} \cdot \begin{vmatrix} a_k \\ b_k \\ c_k \\ d_k \\ e_k \\ f_k \end{vmatrix} > \begin{vmatrix} 1 & t_{j-1} & t_{j-1}^2 & t_{j-1}^3 & t_{j-1}^4 & t_{j-1}^5 \end{vmatrix} \cdot \begin{vmatrix} a_l \\ b_l \\ c_l \\ d_l \\ e_l \\ f_l \end{vmatrix}$$

3.3 Joint smoothness constraints

This constraint is designed to smooth the spline joint. Given the assumption that two segments, seg_k and seg_{k+1}, are connected, and the accumulated **s** of segment is seg_k is s_k, Apollo calculates the constraint equation as:

$$f_k(t_k) = f_{k+1}(t_0)$$

Namely:

$$\begin{vmatrix}1 & t_k & t_k^2 & t_k^3 & t_k^4 & t_k^5\end{vmatrix} \cdot \begin{vmatrix}a_{k0}\\a_{k1}\\a_{k2}\\a_{k3}\\a_{k4}\\a_{k5}\end{vmatrix} = \begin{vmatrix}1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5\end{vmatrix} \cdot \begin{vmatrix}a_{k+1,0}\\a_{k+1,1}\\a_{k+1,2}\\a_{k+1,3}\\a_{k+1,4}\\a_{k+1,5}\end{vmatrix}$$

Then

$$\begin{vmatrix}1 & t_k & t_k^2 & t_k^3 & t_k^4 & t_k^5 & -1 & -t_0 & -t_0^2 & -t_0^3 & -t_0^4 & -t_0^5\end{vmatrix} \cdot \begin{vmatrix}a_{k0}\\a_{k1}\\a_{k2}\\a_{k3}\\a_{k4}\\a_{k5}\\a_{k+1,0}\\a_{k+1,1}\\a_{k+1,2}\\a_{k+1,3}\\a_{k+1,4}\\a_{k+1,5}\end{vmatrix} = 0$$

The result is t_0 = 0 in the equation.

Similarly calculate the equality constraints for

$$\begin{aligned} f_k'(t_k) &= f_{k+1}'(t_0) \\ f_k''(t_k) &= f_{k+1}''(t_0) \\ f_k'''(t_k) &= f_{k+1}'''(t_0) \end{aligned}$$

3.4 Sampled points for boundary constraint

Evenly sample **m** points along the path, and check the obstacle boundary at those points. Convert the constraint into QP inequality constraints, using:

$$Ax \leq b$$

Apollo first finds the lower boundary $l_{lb,j}$ at those points (s_j, l_j) and $j \in [0, m]$ based on the road width and surrounding obstacles. Then it calculates the inequality constraints as:

$$\begin{vmatrix}1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5\\1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5\\\cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\1 & t_m & t_m^2 & t_m^3 & t_m^4 & t_m^5\end{vmatrix} \cdot \begin{vmatrix}a_i\\b_i\\c_i\\d_i\\e_i\\f_i\end{vmatrix} \leq \begin{vmatrix}l_{lb,0}\\l_{lb,1}\\\cdots\\l_{lb,m}\end{vmatrix}$$

Similarly, for upper boundary $l_{ub,j}$, Apollo calculates the inequality constraints as:

$$\begin{vmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & t_m & t_m^2 & t_m^3 & t_m^4 & t_m^5 \end{vmatrix} \cdot \begin{vmatrix} a_i \\ b_i \\ c_i \\ d_i \\ e_i \\ f_i \end{vmatrix} \leq -1 \cdot \begin{vmatrix} l_{ub,0} \\ l_{ub,1} \\ \dots \\ l_{ub,m} \end{vmatrix}$$

3.5 Speed Boundary constraint

Apollo establishes a speed limit boundary as well.

Sample **m** points on the st curve, and get speed limits defined as an upper boundary and a lower boundary for each point j, e.g., vub, j and vlb, j . The constraints are defined as:

$$f'(t_j) \geq v_{lb,j}$$

Namely

$$\begin{vmatrix} 0 & 1 & t_0 & t_0^2 & t_0^3 & t_0^4 \\ 0 & 1 & t_1 & t_1^2 & t_1^3 & t_1^4 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & t_m & t_m^2 & t_m^3 & t_m^4 \end{vmatrix} \cdot \begin{vmatrix} a_i \\ b_i \\ c_i \\ d_i \\ e_i \\ f_i \end{vmatrix} \geq \begin{vmatrix} v_{lb,0} \\ v_{lb,1} \\ \dots \\ v_{lb,m} \end{vmatrix}$$

And

$$f'(t_j) \leq v_{ub,j}$$

Namely

$$\begin{vmatrix} 0 & 1 & t_0 & t_0^2 & t_0^3 & t_0^4 \\ 0 & 1 & t_1 & t_1^2 & t_1^3 & t_1^4 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & t_m & t_m^2 & t_m^3 & t_m^4 \end{vmatrix} \cdot \begin{vmatrix} a_i \\ b_i \\ c_i \\ d_i \\ e_i \\ f_i \end{vmatrix} \leq \begin{vmatrix} v_{ub,0} \\ v_{ub,1} \\ \dots \\ v_{ub,m} \end{vmatrix}$$